Design of an Intelligent Bear Robot with Emotional Learning Capabilities for Enhanced Human-Robot Interaction

Authors: Hyung Ju Suh , Woo-Sik Lee, Ju Hwan Seo

Abstract

In this research a bear robot was made that could emotionally interact with human beings by learning from human behavior and adjusting its reaction accordingly. A series of sensors were used to detect human facial expression, approaching velocity, and behavior upon the bear. While in the learning mode, a classifier is constructed using a support vector machine algorithm, which maps the detected facial expression and approaching velocity to a behavior label. After the classifier is constructed the bear predicts human behavior among itself by reading facial expression and approaching velocity; it accordingly reacts to the human behavior to inform the human whether it is in favor of, or against the predicted human behavior.

1. Introduction

It can be said that the ultimate purpose of artificial intelligence research is to create machine intelligence perfectly analogous to that of human intelligence. For artificial intelligence to be truly intelligent and interact with human beings, however, it is imperative that it understands the concept of emotion; emotion is undoubtedly a large part of motivation for human actions, or interaction among each other. Research regarding machine's emotion has been largely advanced in light of development from cognitive psychology and artificial intelligence. Yet we do not yet know precisely what comprises human emotions, how various components of emotions work, or how we can apply the concept of emotion to machines.

In light of previous research that has been conducted regarding machine emotion, R.W. Picard argues in *Affective Computing* that for a machine system to have emotions, it needs to have five components: emotional behavior, fast primary emotions, cognitively generated emotions, emotional experience, and body-mind interaction. [1] While the other four components can be pre-programmable using a set of algorithms, emotional experience is a factor that must be taught to the robot after the rest of the architecture is set. Hyung-rock Kim, in his research to program the architecture for the other four factors of emotion, noted that "emotional experience is related with highly elaborated functions of emotions,"[2] and did not incorporate experience due to various difficulties.

Indeed emotional experience and the concept of teaching a machine how to 'feel' is extremely difficult since countless events of experience can affect the emotional system differently. This research, however, will attempt to simplify the process by assuming a very simple emotional architecture for the robot, and focusing on how the machine can emotionally learn from human behavior. In order to do this a robot is created in a form of a teddy bear to maximize empathy for emotional interaction from human beings. The robot assumes a very simple emotional architecture purely based on primary stimulus: it performs a negative motion upon strong stimulus and positive motion upon weak stimulus. In easier words, the bear prefers petting and rejects hitting.

Based on this emotional architecture the robot recognizes two emotional indicators from human beings, which are facial expression and approaching velocity. In learning mode direct stimulus to the bear is measured, and the robot maps the emotional indicators to a stimulus label. Then using a support vector machine in two-dimensional space, a classifier is constructed which predicts the stimulus based on emotional indicators. Using such classifier, the bear uses what it has learned to react to human emotion indicators. Using such machine learning methods in emotional interaction could be beneficial due to the difficulties of incorporating emotional experience that were proposed above; countless numbers of experience and its reactions are impossible to program in real-life. Therefore a much more effective solution would be to let the machine learn along the way.

Thus overall this research aims to do the following: first, the research aims to design a bear robot and its scenario that can learn from human behavior to adjust its actions. Next, the research will try to incorporate machine learning into the concept of emotional experience. Finally and ultimately, this research attempts to enhance human-robot interaction by giving the robot the ability to learn human behavior in an emotional context.

2. Scenario Design

For the sake of programming and data collecting, a scenario was designed for the whole plan beforehand. In the research three sensors are used. To detect facial expression, a webcam was used to recognize face. For approaching velocity, a Kinect sensor was used. Finally, instead of measuring direct stimulus via pressure sensors, Leap Motion was used to measure hand velocity. The proposed scenario is written below.

Largely the scheme is divided into a learning mode scenario, and acting mode scenario. In the learning mode, the Kinect sensor first triggers the algorithm by detecting human beings. If a human being has been detected, then the Kinect sensor reads the average approaching velocity from a specific distance to a closer distance. Simultaneously, a webcam detects human facial expression and passes on a value from a specific point. Finally, the leap motion sensor detects the maximum hand velocity. The scenario was for data collection was planned this way because a single time-independent quantity was desired. The data that was collected are saved in a file in a three-dimensional vector (i.e. (Kinect value, webcam value, Leap Motion value)). Utilizing the data achieved from the learning mode, a classifier is made in a support vector machine (SVM) using the leap motion value as a label.

After a classifier is constructed a reaction mode algorithm can be applied. In this mode Kinect detects the person and triggers the algorithm. Then while Kinect measures the average approaching velocity, the webcam simultaneously detects human facial expression. The data is then sent to the classifier; based on the predicted stimuli, the bear robot's action is decided. If Kinect judges that the person is in a sufficiently close distance, the robot carries out the motion. The proposed scenario is summarized below in a diagram. The server sending part is due to socket programming that was used to connect different programs in the project.

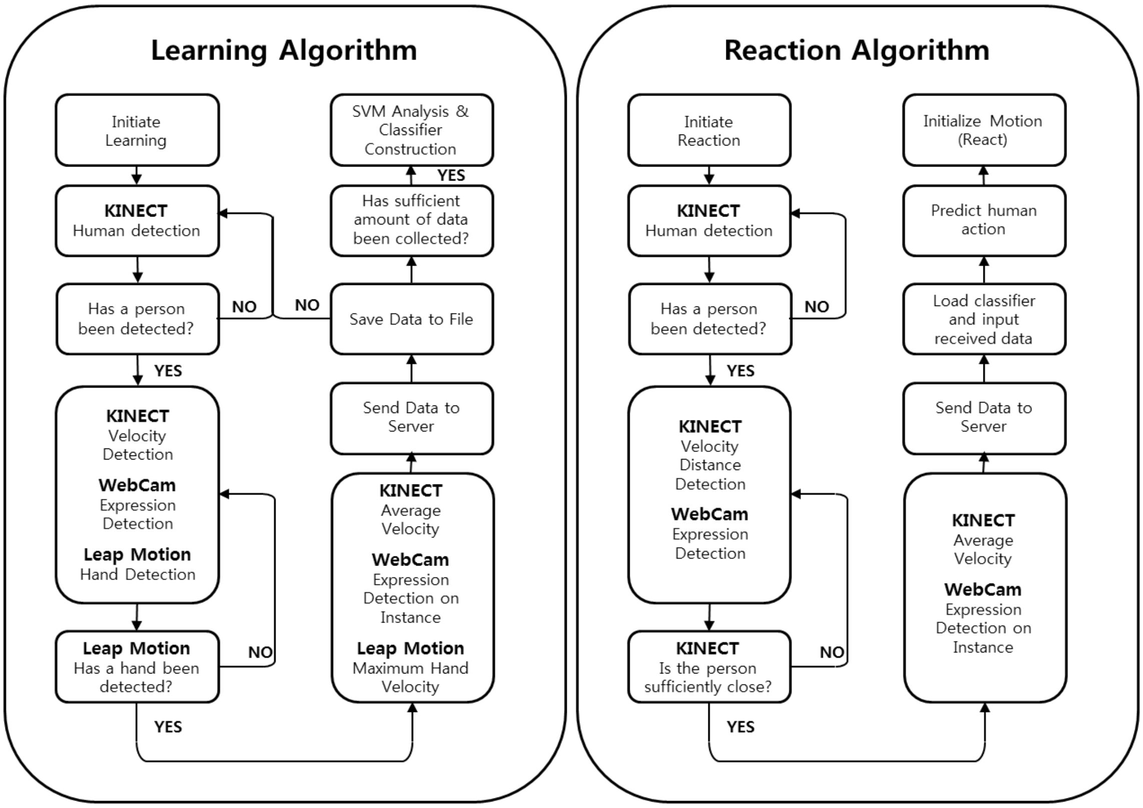

Learning Algorithm
Initiate Learning
KINECT Human detection
Has a person been detected?
NO
YES
KINECT Velocity Detection
WebCam Expression Detection
Leap Motion Hand Detection
Leap Motion Has a hand been detected?
NO
YES
KINECT Average Velocity
WebCam Expression Detection on Instance
Leap Motion Maximum Hand Velocity
Send Data to Server
Save Data to File
NO
Has sufficient amount of data been collected?
YES
SVM Analysis & Classifier Construction
Reaction Algorithm
Initiate Reaction
KINECT Human detection
Has a person been detected?
NO
YES
KINECT Velocity Distance Detection
WebCam Expression Detection
KINECT Is the person sufficiently close?
NO
YES
KINECT Average Velocity
WebCam Expression Detection on Instance
Send Data to Server
Load classifier and input received data
Predict human action
Initialize Motion (React)

Figure 1. Diagram of the designed Scenario

3. Sensors and Programming

3.1 Expression Detection

In the research human expression was used as one of the primary emotion indicators. While facial recognition is readily available as open source, expression recognition was hard to find; thus a facial recognition program was adjusted in order to detect expression. OpenCV 2.4[3] and its internal face recognition algorithm were used in order to detect faces. Then, facial landmarks were detected from the recognized face using a program designed by

M. Uricar et al. [3]. After obtaining the coordinates of the facial landmarks using the program, the ratio between eye-nose distance and nose-mouth distance was used as a factor to quantize facial expression. For convenience, this ratio was termed k-ratio. Below is the result of Uricar's landmark detection program and a diagram explaining in detail the coordinates that were obtained and the k-ratio.

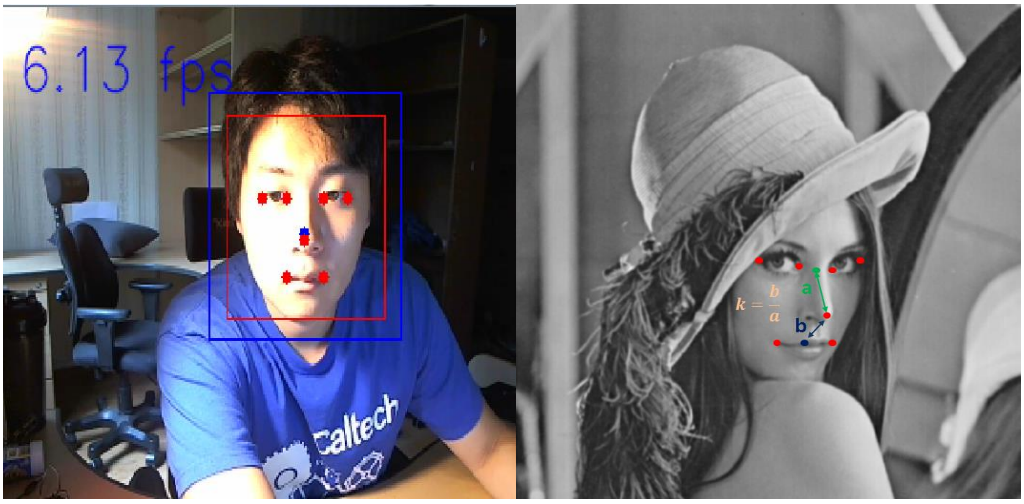
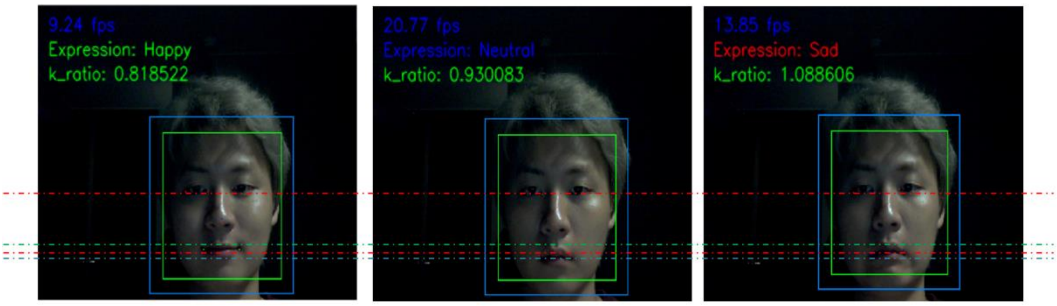


Figure 2. Screenshot of the landmark detection program and the definition of k-ratio

Seven red dots that are placed on the face indicate the coordinates of the face that result from the landmark reading program. Then the midpoint of the eye coordinates and the midpoint of the mouth coordinates are defined. Then the distance between the midpoint of the eye and the nose coordinate works as the denominator, while the distance between the midpoint of the mouth and the nose coordinate becomes the nominator.

Because the distance between the mouth and the nose decreases as a person laughs and increases as a person frowns, the k-ratio can be used as a successful elementary quantity that can decide facial expression. Also, because data acquisition has a lot of fluctuation due to instability of landmark detection, the data goes through an average filter of adjustable size. The result of the modified facial expression detector is the following.



9.24 fps
Expression: Happy
k_ratio: 0.818522

20.77 fps
Expression: Neutral
k_ratio: 0.930083

13.85 fps
Expression: Sad
k_ratio: 1.088606

Figure 3. Screenshot of the modified expression detection program and its performance

As seen in the above, the program is successful in quantizing and detecting facial expressions. However, there exist many restraints to this rather elementary method. Because the proposed k-ratio is different from person to person, the ratio does not provide a universal quantity that applies for every individual. The ratio also suffers from angle constraints as the ratio will be sensitive to the direction at which the person looks at the camera. Finally, the threshold value for judging whether a person is happy, neutral, or sad is subjective and prone to debate. A more universal expression recognition program would improve the research greatly, but as facial recognition is not a central issue of the research, this elementary method was used.

In the scenario, as a person approaches the bear (and thus the camera), the camera continuously reads the distance between facial coordinates and passes on a value at a single instance if the distance between a facial landmark exceeds a threshold value. For this purpose the distance between the two mouth landmarks were utilized as an indicator of how close the person is. This method can be perceived as problematic as a person goes through a series of different facial expressions as he or she approaches the bear. Regardless a single instance method was selected due to the convenience that time-independence holds in machine learning.

3.2 Approaching Velocity

A Kinect sensor was utilized to read another important emotional indicator, approaching velocity. Approaching velocity can be a direct indicator of how aroused a person is; if he is in high velocity the person is angry, or in great joy. On the other hand if she is in low velocity the person is depressed, or content. Using the existing built-in functions for Kinect, a person’s approaching speed and distance was measured. Then the Kinect sensor was programmed to continuously read the speed values and obtaining the average velocity until the person came up to a threshold value. Taking an average is a good indicator of the overall concept of how fast a person is coming. Unlike expression, velocity has a lot of fluctuation and thus taking a single velocity value for a set moment is not favorable. Upon taking an average, however, a person’s intent can be represented without much fluctuation, except for exceptional cases which can be ruled out for the sake of scenario. (For example, when a person suddenly pauses before coming to interact with the robot.) All of the software was programmed using the Kinect development kit from Microsoft [5].

3.3 Leap Motion

Leap Motion was used in order to measure stimulus on the bear. This stimulus later becomes a label for the machine learning process. The initial and intuitive approach to measuring stimulus was using pressure sensors on the bear, but the range of pressures plates and the difficulty of processing the signal became problematic. A leap motion would be a fitting alternative as a hand’s velocity can be said to have direct correlation with the impact it has on the bear. The sensor was programmed so that upon hand detection, it would store the hand’s maximum velocity value. When the hand no longer appears on the sensor, the sensor terminates the process and returns the maximum velocity detected. All of the software was programmed using leap motion development kit [6].

After the velocity is detected, the value is put into one of the three threshold sections: hit, neutral, or pet. Accordingly the bear’s reaction is later programmed in response to such actions. Upon predicting hitting the bear will take defensive motion, while upon petting the bear will take favorable actions. Below is the exact setup for the scenario and the exact location of the sensor while obtaining data and performing the reaction scenario.

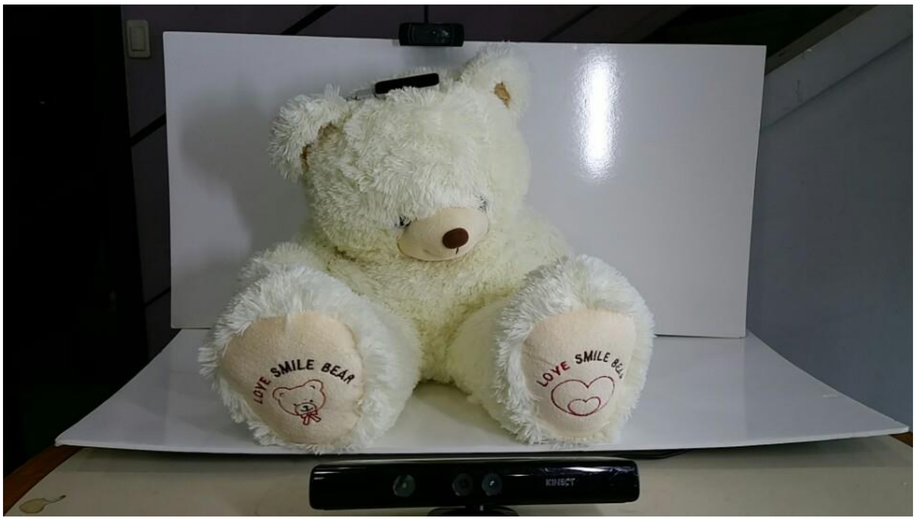


Figure 4. Direct shot of the experiment’s setup while data collection

4. Machine Learning Process

In the machine learning process, a support vector machine (SVM) was chosen as our algorithm for its simplicity and applicability. The designed scenario during the data collection process allows us to store data in a three-dimensional vector, in the format of (Approaching velocity, Facial Expression, Hand Velocity). Then the raw data goes through a data processing filter which calibrates the values for a SVM based on visual representation. Approaching velocity values become the x-coordinate, and facial expression values become the y-coordinate. The hand velocity is thresholded into three regions, and becomes a label for one of the points in the two-dimensional space.

Then the SVM algorithm is applied to these points. In the research OpenCV 2.7's built-in SVM function was utilized to construct the classifier, with the kernel being linear. It can be justifiable to use the linear kernel in this case because the data clearly shows tendency. For example, a strong stimulus will mostly come from sad expression and fast velocity, while a favorable stimulus will come from a happy expression and low velocity. Below is a visual representation for the machine learning process that was carried out. A definite tendency is observable for ten sets of data that were obtained. Here the x-axis represents approaching velocity and y-axis represents facial expression values.

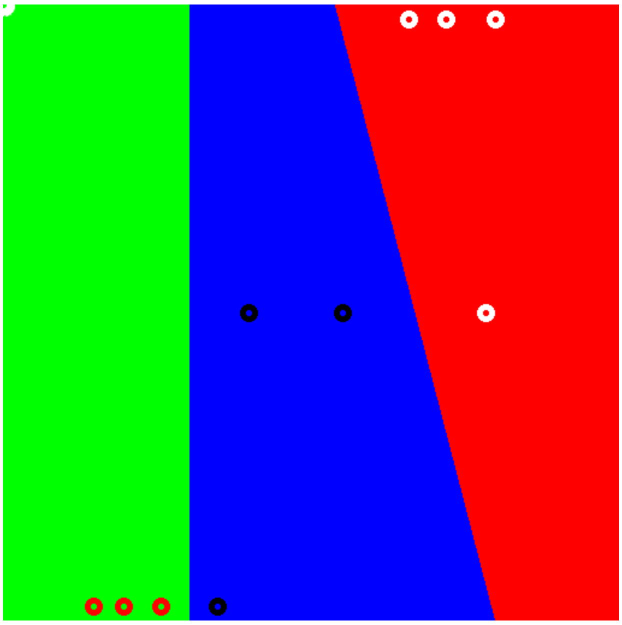


Figure 5. Visual representation of SVM software's performance based on 3 labels.

5. Robot Hardware & Motion Design

In the reaction mode algorithm of the robot, the robot needs to act according to the predicted stimuli. Thus a robot hardware was made to fit inside a bear doll. Eleven Dynamixel servos were used to provide 9 degrees of freedom for actions, and two servos for forward and backward movements. Three degrees of freedom were provided for each of the arms, one for the head, and one for each of the legs. The picture of the resulting robot is provided below.

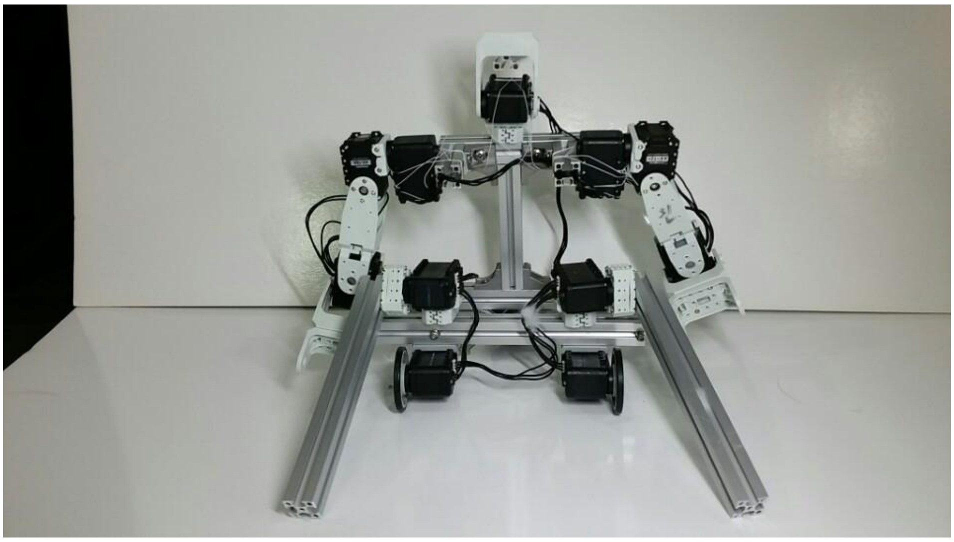


Figure 6. Assembled hardware inside the bear robot using Dynamixel and profile frame.

After designing and assembling the hardware, the motion that is related to each of the stimulus had to be planned. Although motion planning for emotional interaction stands as a large topic on its own, simple motions can be designed based on elementary emotional cues. All of the software was developed using Dr. J. Yang's dynamixel control functions. [7]

For an affirmative motion, the arm servos were programmed to look like the bear is giving a hug to the approaching person, in order to express openness. Also, the head servo was programmed so that the bear would slowly nod and look down, as if it wants to be petted. The legs were set to go up and down slowly. Time-lapse images of the bear's motion are shown below:

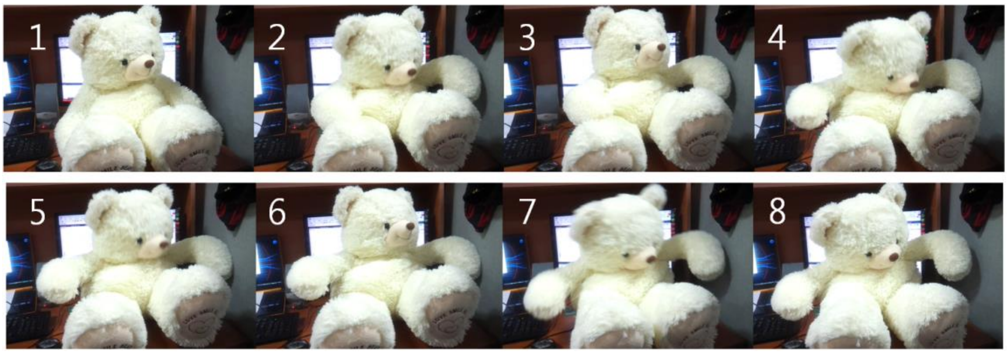


Figure 7. Time-lapse photograph of the bear's favoring motion

Negative action was designed so that the bear looks like it is defending itself, and expressing refusal to the approaching owner. The arms were programmed so that one arm would cover the bear's face to show fear, while the other points away from the user expressing refusal. The head is raised to express alarm, and the legs move comparatively rapidly in order to express fear. Time-lapse images are shown below:

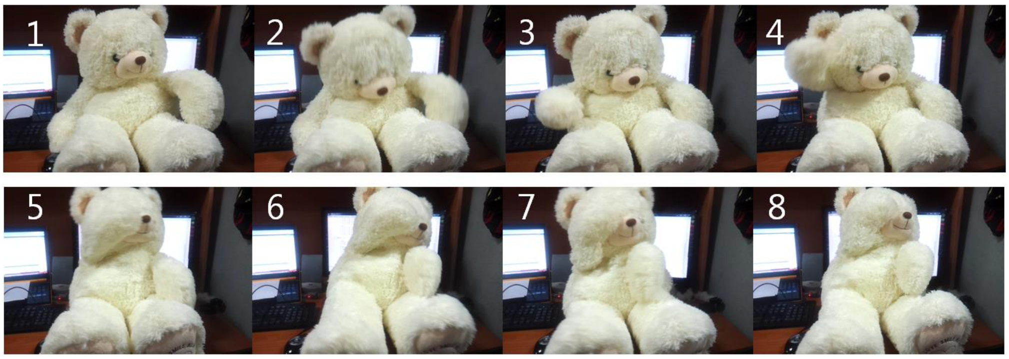


Figure 8. Time-lapse photograph of the bear's rejecting motion

The neutral motion was also planned to show a moderately medium and ambiguous motion between the two reactions. The arm simply moves vertically so that no emotion is correlated to it, while the head simply nods in normal speed.

6. Results and Demonstration

After checking that each of the programs is running correctly, socket programming was used to communicate data between each program. The involved programs were related Kinect, Leap Motion, Expression Detection, SVM, and motion planning. In learning mode, 20 data sets were retrieved by recording the facial expression, approaching velocity, and hand velocity of each instance. Below is an example of the training process.

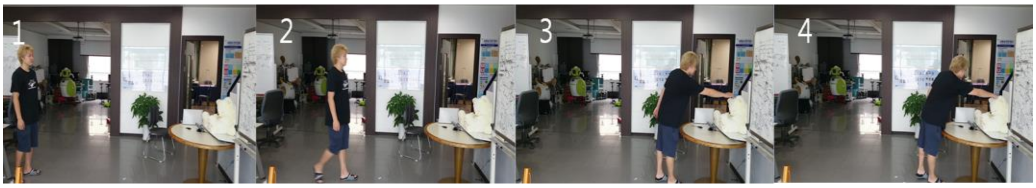


Figure 9. Time-lapse photograph of the learning environment demonstration with positive stimuli

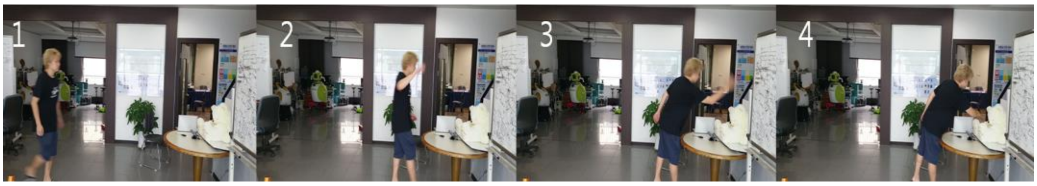


Figure 10. Time-lapse photograph of the learning environment demonstration with negative stimuli

After the training process, the received data was analyzed by a SVM algorithm to create a classifier, and the classifier was connected to the robot's motion so that the robot would predict a man-given stimulus based on the person's expression and approaching speed. The planned scenario was applied when designing the demonstration.

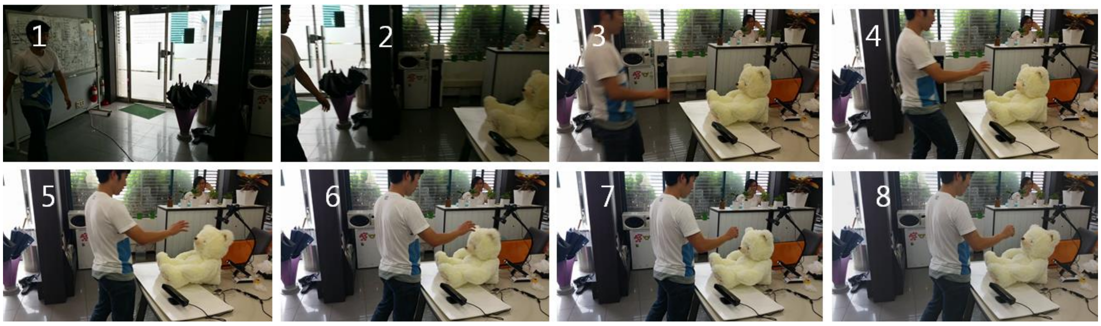


Figure 11. Time-lapse photograph of the acting environment demonstration with positive predicted stimuli

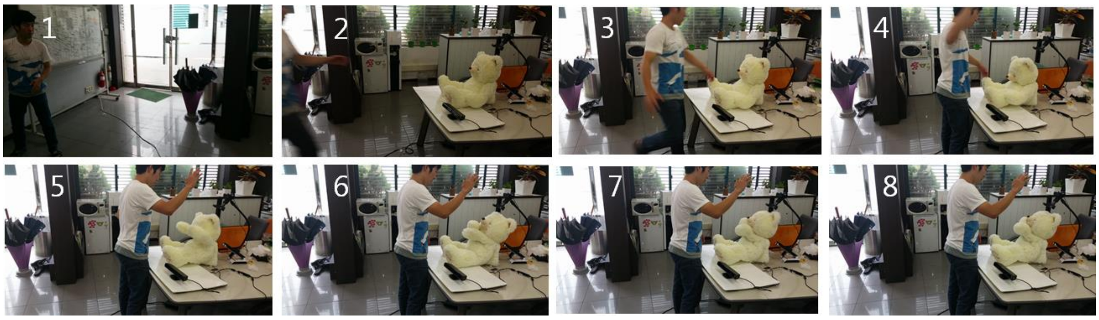


Figure 12. Time-lapse photograph of the acting environment demonstration with negative predicted stimuli

7. Conclusion and Future Works

Returning to the first aim of the research, the bear robot and its scenario was successfully designed, using a set of sensors, SVM, and servo hardware, to learn from human behavior to adjust its actions accordingly. In learning mode the bear is capable of analyzing the intent of a person by experiencing the actions that the person takes upon two emotional indicators: approaching velocity and facial expression. In reaction mode the bear can form a classifier using machine learning based on the acquired data during learning mode. Then the bear robot can adjust its reaction based on the classifier that is formed from its experience. The scenario and the programmed is successful in meeting the intent of the research.

The next aim of the research was to incorporate machine learning algorithms in research regarding robot emotions. Since the robot can use an SVM to analyze the user's intent and predict it based on emotional cues, machine learning in this case successfully enables the robot to react accordingly. Although performing the same task by dividing cases and programming for every case would not prove much more troublesome due to the small number of emotional indicators and motions in this research, machine learning would prove greatly effective were indicators and cases became much more diverse.

In the end, we can also see that giving the bear robot emotional learning capability effectively enhances human-robot interaction as the robot learns how different degrees of emotional cues connect to different human emotions based on its emotional architecture. Such learning capability also enhances interaction by enabling customizability for robots designed for emotional empathy. A carefully designed machine-learning system for emotional interaction will detect different degrees of emotional cues for different people and make it possible to adjust the robot's reaction accordingly.

Because this research is comprehensive among many different fields of research, much room for improvement and future topics exist within the research. For real-world applications, much various emotional cues and cases exist such as voice, gesture, or a more detailed expression cue. Incorporating these emotional cues would require much more elaborate function to connect them, and a far more comprehensive algorithm in order to apply machine learning. The sensors could also be better built within the research. For example, facial expression detection is a large field of research on its own and the elementary method that was used in the research could be reinforced. Motion design is also a part that could be improved and researched on to improve the bear robot. Finally, the simple emotional architecture that was assumed in the researched could be improved to contain a more realistic one. Learning emotions based the proposed behavior would be much more complex in this case, since functions within the architecture would incorporated.

8. References

[1] R.W. Picard, *Affective Computing*. Cambridge, Massachusetts: The MIT Press, 1997.

[2] Hyoung-Rock Kim, *Hybrid Emotion Generation Architecture with Computational Models Based on Psychological Theory for Human-Robot Interaction*, Ph.D. Thesis 2009, KAIST, Republic of Korea

[3] Bradski, G. (2000) Dr.Dobb's Journal of Software Tools, 2008

[4] M. Uricar, V. Franc and V. Hlavac, *Detector of Facial Landmarks Learned by the Structured Output SVM*, VISAPP 2012: Proceedings of the 7th International Conference on Computer Vision Theory and Applications, 2012

[5] Microsoft Kinect SDK 2.0, 2014, http://www.microsoft.com/en-us/kinectforwindows/develop/

[6] Leap Motion SDK v2, https://developer.leapmotion.com

[7] J. Yang, 2012, Dynamixel Control Software, Telerobotics and Controls Laboratory, KAIST, Republic of Korea